

## Contenido

INTRODUCCIÓN.....	1
SUBCONSULTAS CON LOS OPERADORES: >, <, >=, <=, !=, =.....	2
Ejercicios: .....	3
SUBCONSULTAS CON IN Y NOT IN .....	3
Ejercicios: .....	4
SUBCONSULTAS CON AÑO o AJO.....	5
Ejercicios: .....	6

## INTRODUCCIÓN

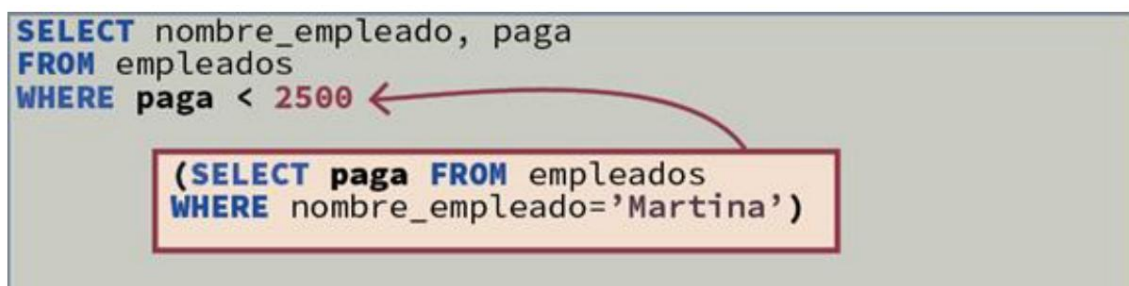
En ocasiones, es necesario ejecutar una sentencia SELECT para conseguir un resultado que utilizar como parte de una condición en otra sentencia SELECT. El lenguaje SQL nos facilita efectuar este tipo de operaciones utilizando las subconsultas.

Una subconsulta es una sentencia SELECT que se incluye formando parte de una condición en la cláusula WHERE o en la cláusula HAVING de otra sentencia SELECT.

La subconsulta está entre paréntesis.

Nota: Una subconsulta puede contener, al mismo tiempo, otras subconsultas (nosotros sólo veremos un primer nivel de implantación).

En realidad, lo primero que hace la base de datos es calcular el resultado de la subconsulta:



```
SELECT nombre_empleado, paga
FROM empleados
WHERE paga < 2500

(SELECT paga FROM empleados
WHERE nombre_empleado='Martina')
```

La consulta obtiene a los empleados cuyas pagas sean menores que el que gana martina (2500) .

Las subconsultas siempre deben cerrarse entre paréntesis y se deberían (aunque no es obligatorio, sí altamente recomendable) colocar a la derecha del operador relacional.

## SUBCONSULTAS CON LOS OPERADORES: >, <, >=, <=, !=, =

La sintaxis sería la siguiente:

```
SELECT listaExpresiones  
FROM mesa o tablas  
WHERE expresión OPERADOR  
  
      (SELECT que devuelve un ÚNICO VALOR);
```

Cuando nosotros usamos uno de estos operadores en una condición comparamos la expresión con UN VALOR. De ahí la importancia de que la SUBSELECT devuelva en este caso una única fila con una única proyección.

Por ejemplo: Datos de los Productos que tengan un precio mínimo.

```
SELECCIONAR *  
  
FROM Productos  
  
DONDE PrecioVenta =(SELECCIONAR MIN(PrecioVenta)  
  
                        FROM Productos)
```

Como ve en este ejemplo, la subconsulta (SELECT MIN(PrecioVenta)  
FROM Productos) devuelve un único valor.

También podríamos usar subconsultas en la cláusula HAVING como parte de una condición para a los grupos. La sintaxis sería la siguiente:

```
SELECT listaExpresiones  
FROM mesa o tablas  
  
[WHERE condiciones]  
  
Criterios de AGRUPAR POR  
  
HAVING expresión OPERADOR  
  
      (SELECT que devuelve un ÚNICO VALOR);
```

Por ejemplo: Gamas de Productos que tienen un precio medio por encima del precio medio de los Productos. Proyectar: precio medio, máximo y mínimo de venta.

El precio medio de los Productos (la subconsulta) lo obtendríamos:

```
SELECCIONAR AVG(PrecioVenta)

FROM Productos
```

Ahora hacemos la proyección que nos piden, pero no queremos ver a todos los grupos que se nos forman sino sólo aquellos que cumplen la condición:

```
SELECT Gama, AVG(PrecioVenta), MAX(PrecioVenta), MIN(PrecioVenta)

FROM Productos

GRUPO POR Gama

TENIENDO AVG(PrecioVenta) > ( SELECCIONAR AVG(PrecioVenta)

                                FROM Productos)
```

---

#### Ejercicios:

1. Datos de las Oficinas ubicadas en el país del cliente de CódigoCliente 7
2. Datos de los productos que tienen el precio inferior al precio medio de los productos de la gama 'Ornamentales'.
3. Datos de los pedidos que tienen menos productos que el pedido de CódigoPedido 1.
4. Datos de los pedidos realizados en una fecha anterior al pedido de código 14.
5. Datos de los pedidos realizados en un año distintos al pedido de Código 1.
6. CódigoPedido e importe de los pedidos que tiene un importe mayor al pedido de Código 1.
7. Ciudades que tienen más clientes que 'London'.

---

## SUBCONSULTAS CON IN Y NOT IN


En el primer boletín ya vimos cómo funcionaban IN y NOT IN con literales.

IN comparaba una expresión con una lista de valores; si el valor de la expresión era igual al de algún elemento de la lista, el resultado de la condición era VERDADERO.


La lista, como vimos, pueden ser literales o podemos generar la lista a partir de una subconsulta. Como podrá deducir en la subconsulta sólo se puede proyectar en SELECT una columna.

La subconsulta debe generar una lista de valores compatibles con la expresión1. Por tanto, en el SELECT de la subconsulta SÓLO PODRÁ HABER UNA COLUMNA. En otro caso, nos generaría parejas de valores o ternas de valores...lo cual no es comparable con el valor de una expresión.

La sintaxis sería la siguiente:

```
SELECT listaExpresiones
FROM mesa o tablas
DONDE expresión1 EN 
                     (SELECT que retorna una ÚNICA COLUMNA)
```

el

```
SELECT listaExpresiones
FROM mesa o tablas
[WHERE condiciones]
Criterios de AGRUPAR POR
HAVING expresión IN 
                    (SELECT que retorna una ÚNICA COLUMNA)
```

Por supuesto, la misma sintaxis sería para NOT IN.

NOT IN dará VERDADERO cuando la expresión no coincida con ningún valor de la lista.

Nota: Fíjese en que todas las condiciones que se pueden hacer con los operadores = o != se pueden realizarse con IN o NOT IN respectivamente. Una lista puede constituir un solo elemento.

---

#### Ejercicios:

8. Datos de los Clientes ubicados en ciudades donde hay alguna oficina.
9. Datos de los Pedidos realizados por clientes que han realizado algún pago.
10. Datos de los Productos cuyo precio de venta no coincide con el de ningún producto de la gama 'Herramientas'
11. Datos de los Pedidos que no tengan entre sus productos productos de la gama 'Herramientas'.  
(Pista: ¿Podemos averiguar el CódigoPedido de los pedidos que tengan alguna herramienta?...Pues esos no)

12. CodigoCliente y Fecha de su primer pedido de los clientes que hayan hecho algún pedido en el 2018
13. Pedidos que se realizaron en la fecha en que se realizó algún pago
- 

## SUBCONSULTAS CON AÑO o AJO

Con IN o NOT IN sólo podemos realizar comparaciones con subconsultas de igualdad o desigualdad respectivamente. Cuando necesitamos que la comparación con los elementos de la lista sea otra (>, <, >=, <=) necesitamos usar ALL o AÑO

CUALQUIER el ALGUNO	Compara con cualquier valor de la subconsulta. La condición se evaluará a verdadero si la comparación es una cierta con al menos uno de los valores que devuelve la subconsulta. Se suele utilizar la palabra AÑO (SOME es un sinónimo)
TODO	Compara con todos los valores de la subconsulta. La instrucción resulta cierta si es cierta la comparación para todos los valores de la subconsulta

La sintaxis será la siguiente:

```
SELECT listaExpresiones
FROM mesa o tablas
WHERE expresión operador AÑO
(SELECT que retorna una ÚNICA COLUMNA)
```

el

```
SELECT listaExpresiones
FROM mesa o tablas
[WHERE condiciones]
Criterios de AGRUPAR POR
HAVING expresión operador AÑO
(SELECT que retorna una ÚNICA COLUMNA)
```

Obviamente, la sintaxis será igual para ALL.

El OPERADOR puede ser cualquier operador de comparación. Así tendremos las siguientes equivalencias:

EN	= CUALQUIERA
NO EN	=! TODOS

Por ejemplo: Si queremos saber los datos de los Productos que tienen un precio mayor al de cualquier producto de la gama 'Herramientas'. Esto significa que el precio de los Productos seleccionados deberá ser mayor al de TODAS las herramientas.

★  
SELECCIONAR

FROM Productos

DONDE PrecioVenta > TODOS( SELECCIONAR PrecioVenta

FROM Productos

WHERE Gama like 'Herramientas')

Hay que tener cuidado cuando la consulta tiene un enunciado de " el más..." o "el menos..." dado que el valor máximo o el valor mínimo puede estar incluido en la lista generada por la subconsulta.

Por ejemplo: Datos del Producto más caro. El producto (o Productos: podría haber varios con el precio máximo) más caro será el que su precio mayor o igual para los elementos de una lista formada por los precios de todos los Productos (incluido el mismo)

★  
SELECCIONAR

FROM Productos

DONDE PrecioVenta >= ALL( SELECT PrecioVenta

FROM Productos)

#### Ejercicios:

14. Precio de los productos que tienen un precio de venta mayor al del precio medio de cualquier gama
15. Nombre y PrecioVenta de los Productos que no se han pedido nunca.
16. Datos del pago de mayor cantidad.
17. Datos del pago de mayor cantidad de 2018.
18. Datos del empleado que tiene más clientes
19. Datos de la Gama que tiene menos productos
20. Datos del pedido más barato
21. Año(s) en que se realizaron más pedidos
22. Datos del empleado que tiene más subordinados
23. Productos que tengan un precio único (distinto a cualquier otro producto)